End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication

Jorge Granjal, Edmundo Monteiro, Jorge Sá Silva
University of Coimbra, Portugal
{jgranjal,edmundo,sasilva}@dei.uc.pt

***Abstract*-The Internet of Things (IoT) describes a vision of a future Internet where constrained sensing and actuating devices are part of distributed applications and required to support standard Internet communications with more powerful devices or Internet hosts. This vision will require appropriate end-to-end communications and security mechanisms that are well suited to the constraints and characteristics of sensing devices and applications, while being able to support standard Internet communication mechanisms. With this motivation in mind, we propose an architecture supporting low-power end-to-end transport-layer secure communications with mutual authentication using ECC public-key cryptography for Internet-integrated sensing applications. The proposed architecture promotes the availability of critical resources on constrained sensing platforms and security against Internet-originated threats, while providing full compatibility with current standardization proposals. Those are fundamental enabling factors of most of the sensing applications envisioned for the IoT and, as far as we known, ours is the first architecture implemented and experimentally evaluated with such goals.**

Keywords-Internet of Things, CoAP, DTLS, mutual authentication, delegated ECC public-key authentication

I. INTRODUCTION

Many of the applications currently envisioned for the Internet of Things (IoT) are critical in respect to security, being it security of its users, of the processed data or of the communications. Despite this fact, such applications will interact with physical phenomena by employing very constrained sensing platforms and low-energy wireless communications, aspects that seriously complicate the design and adoption of appropriate security mechanisms. As wireless sensor networks (WSN) applications are starting to require interconnection with the Internet at some degree, end-to-end communications between constrained sensing devices and other Internet entities will be a fundamental requirement of many sensing applications. The support of end-to-end security involving constrained sensing devices will represent a fundamental enabling factor of many IoT applications, as it may provide security even when the underlying network infrastructure is only partially under the user's control. As with protocols such as TLS that play a fundamental role in providing security to applications, end-to-end security at the transport-layer may provide an important contribution to the achievement of appropriate security with Internet-integrated sensor networks.

The constraints in terms of fundamental resources such as memory, microprocessor and energy determine the usage of low-energy wireless communications, providing low communication speeds and small packets with the goal of minimizing communication errors. The integration of low-energy personal area networks (LoWPAN) with the Internet brings new challenges into the design of communication and security mechanisms able to support end-to-end communications between devices that are very different in the support of such capabilities.

Of particular relevance to the adoption of a future communications architecture supporting the integration of LoWPANs with the Internet are the technologies currently being designed and adopted at the IETF, in particular at the IPv6 over Low Power Personal Area Networks (6LoWPAN) [1][2][3] and Constrained RESTful Environments (CoRE) [4][5] working groups. 6LoWPAN provides an adaptation layer enabling the transmission of IPv6 packets over constrained low-energy communication environments, in particular using IEEE 802.15.4 [6] at the physical and media access control layers. The CoRE working group is currently designing the Constrained Application Protocol (CoAP) to support RESTful web communications on similar environments.

Although 6LoWPAN and CoRE provide the mechanisms required for the support of end-to-end communications with Internet-integrated sensing devices, appropriate security mechanisms will be required considering the limitations of such devices and the threats that will arise due to the exposure of LoWPAN environments to Internet communications. Although numerous proposals exist to address security in closed LoWPAN environments [7], the integration of sensor networks with the Internet will raise challenges yet to be faced by research. From a standardization standpoint, the current proposal for the support of transport-layer security on 6LoWPAN environments adopts the DTLS [8] protocol to provide confidentiality, integrity and authentication to CoAP application-layer communications.

While the overhead introduced by DTLS on 6LoWPAN communications is certainly non-negligible, its applicability will be fundamentally dependent on the viability of supporting the security modes currently proposed for CoAP security [4] using constrained sensing platforms. In particular, the impact of Elliptic Curve Cryptography (ECC) must be carefully evaluated, and the same may be applied to the impact of communications related with authentication and key agreement in the context of the DTLS initial handshake. In this context, we propose and experimentally evaluate an architecture enabling security at the transport layer supporting DTLS security as proposed for CoAP, while addressing the

previously discussed issues. Our architecture integrates mechanisms designed to contribute to the effectiveness of end-to-end transport-layer security and to the protection of low-energy wireless communication environments against Internet-originated threats. As far as we known, ours is the first proposal targeting such goals.

Our paper proceeds as follows. Section II analyses related work and Section III discusses the usage of end-to-end security in the context of 6LoWPAN and CoAP communications. The proposed architecture is described in Section IV, and Section V discusses our experimental evaluation study of the proposed mechanisms. Finally, Section VI concludes the paper.

II. Related work

Although new mechanisms will be required to support security with end-to-end communications using recently standardized technologies such as 6LoWPAN and CoAP, particularly considering that such communications may take place in the context of Internet-integrated sensing applications, most of the existing proposals to protect LoWPAN communications target the link-layer and closed LoWPAN environments [7]. In such proposals sensing devices may communicate securely using individual, group or network-wide symmetric encryption keys. For example, MiniSec [9] falls on this category and supports encryption and authentication for unicast and broadcast communications at the link-layer. Regarding the support of security proposals in the context of Internet-integrated LoWPAN environments, fewer research proposals do exist with similar goals as ours. One such proposal is Sizzle [10], implementing a compact web server providing HTTP accesses protected by SSL using 160-bit ECC keys for authentication and key negotiation. Nevertheless, Sizzle requires a reliable transport-layer protocol and is therefore incompatible with CoAP and 6LoWPAN, while also impacting largely in the performance of low-energy communications. Sizzle also only supports authentication of the sensing device but not of the Internet host, while many M2M applications on the IoT are likely to require two-way authentication, as we consider for our proposed architecture. On the other end the SSNAIL [11] proposal supports two-way authentication using an ECC-enabled handshake, but also requires a reliable protocol at the transport-layer. Thus, SSNAIL is also incompatible with 6LoWPAN and CoAP.

Regarding the support of DTLS on constrained 6LoWPAN environments, in [12] the authors propose the compression of DTLS headers with the goal of saving payload space and in consequence reducing the communications overhead. Although DTLS header compression may be of interest, appropriate mapping mechanisms would be required at the border router of an Internet-integrated LoWPAN, or in alternative Internet hosts would be required to support DTLS in its compressed form. Also, this proposal does not address the computational and energetic impact of DTLS authentication and key agreement, certainly a significant part of the whole overhead of DTLS. On the other end, the architecture proposed in [13] supports two-way authentication with DTLS for end-to-end communications with constrained sensing devices, but using devices required to employ specialized trusted-platform modules (TPM) supporting hardware-assisted RSA cryptography and the secure storage of private keys. It doesn't support ECC public-key authentication or public-key cryptography for mainstream devices without a TPM module, also being incompatible with CoAP security [4]. Other aspect we may note is that the two previous proposals do not address the support of transport-layer security in tandem with other security mechanisms designed to protect constrained sensing devices and low-energy communications from Internet-originated threats and attacks. We may envision this to be an important enabling factor of many sensing applications that will require the usage of constrained LoWPAN devices exposed to Internet communications.

The design of an architecture supporting end-to-end security for Internet-integrated sensing applications provides the opportunity to address the previously identified limitations. CoAP security [4] envisions the usage of ECC cryptography, and as such ECC public-key authentication and key negotiation in the context of DTLS is a requirement. In this context, it is important to note that sensing platforms may not be ready to viably support ECC at this stage, as is verified for example in the experimental evaluation study described in [14]. A related limitation is that it may be costly to store and interpret certificates and ECC public-keys in constrained sensing devices with very limited amounts of RAM and ROM memory. Other goal we may address is to leverage security by designing and supporting mechanisms to be employed side-by-side with end-to-end transport-layer security. For example, mechanisms may be required to support control of accesses to resources available on CoAP constrained sensing devices. Related mechanisms may also be necessary supporting operations such as authentication and trust management between devices on the LoWPAN. We may thus consider that the employment of such mechanisms in parallel with transport-layer security may provide an opportunity to promote security as an enabling factor of Internet-integrated sensing applications.

III. End-to-end security using 6LoWPAN and CoAP

The current CoAP proposal [4] enables RESTful web communications on 6LoWPAN environments and defines bindings for the usage of DTLS at the transport layer. Payload space is a scarce resource in IEEE 802.15.4 environments, and consequently header and address compression is prevalent in 6LoWPAN and CoAP specifications. IEEE 802.15.4 provides 127-bytes of total payload space, from which 25 bytes are used for link-layer addressing, thus providing 102-bytes of payload space at the 6LoWPAN adaptation layer. In Figure 1 we illustrate the usage and availability of payload space in IEEE 802.15.4 low-energy communication environments using 6LoWPAN and CoAP when supporting end-to-end communications with Internet hosts. 6LoWPAN IPHC shared-context header compression [3] enables the compression of the UDP/IPv6 header down to 10 bytes, while CoAP requires 4 bytes and DTLS a total of 13 bytes, not considering the space

required for the transportation of security-related data as an Initialization Vector (IV) or authentication (HMAC) fields.

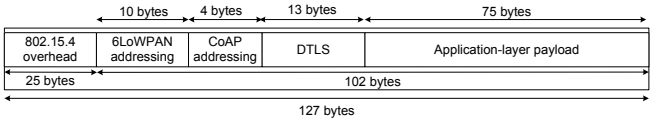

Figure 1. Payload space usage for end-to-end communications in 6LoWPAN environments

Sensing platforms as the TelosB [15] implement IEEE 802.15.4 and support hardware AES/CCM encryption at the link layer. While end-to-end security may dispense link-layer security, this doesn't preclude the usage of hardware-based encryption to support security at higher layers, as we employ in our architecture. While DTLS provides confidentiality, authentication and integrity, the authentication and key agreement between communication parties may take place following different approaches. Three security modes are currently proposed for CoAP with different authentication and key agreement approaches [4]. In the *PreSharedKey* mode a device stores preconfigured keys required to communicate securely with another devices or a group of devices. In the *RawPublicKey* mode a device possesses one or more public keys from where it derives its identification. The third and most interesting mode from the point of view of the integration of LoWPANs with the Internet is the *Certificates* mode, where a device obtains the required public-keys from a certification authority. The later two security modes must employ ECC public-key authentication, in particular authentication of devices and messages using the Elliptic Curve Digital Signature Algorithm (ECDSA) [16] and key agreement using the Elliptic Curve Diffie–Hellman with Ephemeral Keying Algorithm (ECDHE) [16]. Encryption employs AES in CCM (at the hardware when available) or CBC modes.

After authentication, both parties share a pre-master shared secret from which they derive a shared master secret, and from this master secret they obtain the keying material required for encryption and authentication [8]. In terms of security and also of the availability of critical resources, a chain is only as strong as the weakest link. CoAP encryption and authentication using DTLS may be efficiently supported by AES/CCM at the hardware in any of the previously described security modes, but authentication and key agreement may provide the largest impact on the limited resources of low-energy devices and communications, as we discuss next.

IV. End-to-end transport-layer security with mutual and delegated public-key authentication

Although CoAP adopts ECC cryptography in supporting authentication and key negotiation, ECC still represents a non-negligible impact on current sensing platforms as observed in [14]. This limitation is also expressed in the adoption of RSA in proposals such as [13] as an alternative approach. Even though sensing devices may be expected to evolve to support more memory space and increased computational capability in the future, the integration of sensor networks with the Internet must be supported in the near future by mechanisms designed in a realistic fashion, accordingly to the limitations and characteristics of current sensing platforms. One major goal of our work is thus to target alternative approaches for the support of ECC-based public-key authentication and key agreement using "off-the-shelf" sensing platforms, as mechanisms found to be viable for such platforms may be appropriate to a wide range of sensing platforms likely to support future IoT applications. Of particular importance is the overhead of the DTLS handshake and the security of Internet-integrated LoWPAN from Internet-originated threats, two issues that are not addressed in the current 6LoWPAN and CoAP specifications. More specifically, the following are the main concerns addressed in the context of the proposed architecture:

- Overhead of the DTLS authentication and key agreement handshake: other that the payload space required for the DTLS header (around 11% of the available space using 6LoWPAN and CoAP), end-to-end authentication using ECC public-key cryptography requires the exchange of various large messages and certificates. Large handshake messages such as those transporting certificates require fragmentation at the 6LoWPAN adaptation layer. In fact, the most computationally expensive part of a DTLS session is the handshake and it requires more effort from the server than from the client. It is also important to note that many sensing applications are likely to require that sensing devices support CoAP servers. Adding to the time required exchanging handshake messages in low-energy wireless networks at low speeds, sensing devices are required to support ECC public-key authentication and key negotiation. The memory required to store ECC certificates and public-keys might also be a problem, depending on the sensing device and application at hand.

- Protection of end-to-end communications and of sensing devices against Internet-originated threats: DTLS supports limited protection against Denial of Service (DoS) attacks by requiring that a connecting client answers a challenge from the server with a particular stateless cookie. Although this is a desirable mechanism, it may again impact on the resources available on constrained sensing devices. A plethora of similar threats are likely to appear from the minute we start integrating LoWPANs with the Internet.

- Support of ECC cryptography by constrained sensing platforms: mainstream sensing platforms such as the TelosB [15] are unable to efficiently support ECC encryption. This implies that the energy and the computational time required supporting ECC public-key authentication and key agreement undesirably impacts on the lifetime of sensing applications or on its maximum achievable communications rate. Despite such limitations, the support of ECC cryptography in a fashion compatible with the current CoAP proposal is fundamental.

These issues motivate our design of an architecture able to support end-to-end security at the transport-layer for Internet-integrated sensing applications, as we discuss next.

Delegated mutual authentication and key negotiation

The proposed architecture is illustrated in Figure 2, and we consider that a constrained sensing device and an Internet host may both assume the role of the CoAP client or server. The architecture supports end-to-end security at the transport layer for communications between constrained sensing devices and Internet host, with the DTLS handshake being mediated by a 6LoWPAN border router (6LBR). The 6LBR intercepts and forward packets at the transport-layer, a operation that is feasible in the context of its usage as a router supporting communications between the LoWPAN and Internet domains. The computational load related with ECC public-key authentication and key negotiation is thus delegated to the 6LBR, a device we assume without the resource limitations of the CoAP sensor.

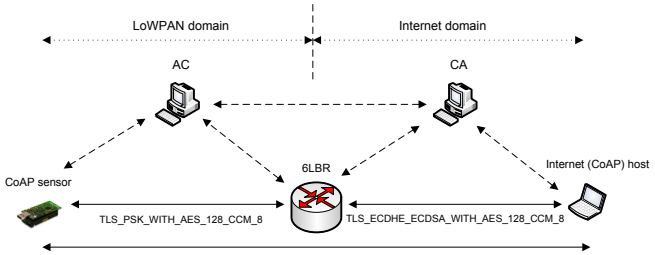

Figure 2. System architecture for end-to-end security via 6LBR

Two other components play important roles in our architecture in the support of authentication and key negotiation. The Certification Authority (CA) server supports ECC public-key certification of communicating entities with X.509 certificates. The Access Control (AC) server supports authentication and trust operations between the 6LBR and sensing devices, as required for the delegation of authentication and key agreement in a secure fashion. This server also provides access control and authorization of secure accesses to CoAP resources, either residing on a CoAP sensing device or on the outside of the LoWPAN (in particular on the Internet).

While guaranteeing end-to-end security, we employ two separate cipher suites for authentication and key negotiation purposes with the two ends of communications. This strategy enables the 6LBR to mediate authentication and key negotiation between both ends while guaranteeing that they end up using the same keying material for end-to-end DTLS encryption and integrity after the initial authentication phase. From the point of view of an Internet host, the 6LBR supports negotiation via the *Certificates* CoAP security mode using the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 cipher suite. The participation of the 6LBR in the authentication and key negotiation phase is transparent to the Internet CoAP device, which is unaware of its presence. On the LoWPAN, the session is negotiated with the CoAP constrained sensing device using the *PreSharedKey* security mode and the corresponding TLS_PSK_WITH_AES_128_CCM_8 cipher suite. This is also transparent to the CoAP sensing device, which is unaware of the fact that it is authenticating via a 6LBR. Thus, while end-to-end security may be achieved supporting the most secure CoAP security mode, on the LoWPAN we make use of a security mode more in line with the capabilities of current sensing platforms. TLS_PSK_WITH_AES_128_CCM_8 may be considered to be the most appropriate cipher suite for LoWPAN environments using devices with the characteristics of the TelosB [15], as authentication and initial key agreement may be performed based on pre-shared secret keys.

End-to-end encryption and integrity using DTLS is supported by AES/CCM after the handshake, and as such our architecture must guarantee that both ends of the communications session use the same keying material. Other goal of the architecture is to support mutual authentication between CoAP endpoints. Contrary to proposals such as [9][10][13], we support mutual authentication over standard 6LoWPAN communications and without requiring the usage of special purpose hardware.

Two-phase mutual DTLS handshake

The first major mechanism of the proposed architecture implements a mediated DTLS handshake supporting delegated ECC public-key authentication. DTLS handshake messages are transparently intercepted by the 6LBR and the handshake is implemented in two phases, with the 6LBR controlling the handshake and supporting ECC cryptographic operations on behalf of CoAP constrained sensing devices. The mediated DTLS handshake employed in our architecture is illustrated in Figure 3.

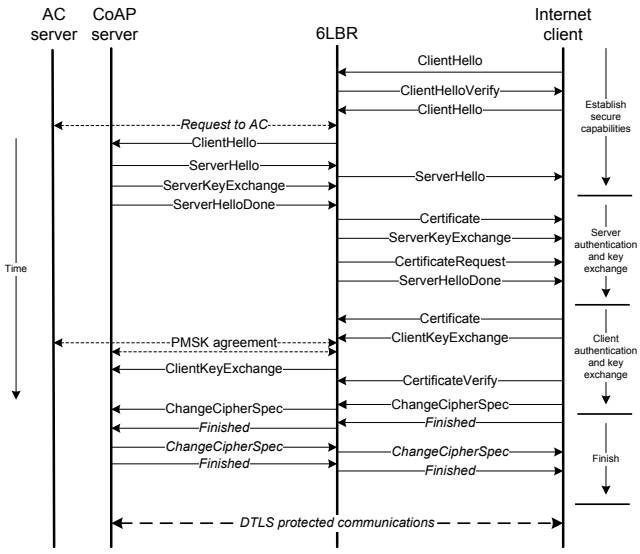

Figure 3. DTLS handshake mediated by a 6LBR

While in the scenario illustrated a CoAP Internet client establishes a secure communication session with a CoAP server residing in a sensing device, the opposite scenario is also supported by this handshake. Figure 3 also illustrates the role

of the AC server in the handshake in supporting authentication of LoWPAN devices. The initial request transported by a *ClientHello* message is transparently intercepted by the 6LBR, which responds with a *ClientHelloVerify* message. This message enables security against DoS attacks at the transport-layer and contains a cookie generated by the 6LBR [8]. The client is required to respond with the same cookie thus proving its willingness to communicate and establish a communication session. The delegation of this mechanism to the 6LBR enables the saving of resources and the protection of the CoAP device against the processing of fake requests.

A secure DTLS session requires the two parties to agree on the cipher suite and encryption keys employed. The handshake supports the transportation of the information required to obtain such secret material. The encryption keys are obtained from a master key that the client and server must share [8]. This master key may, on the other end, be obtained by both parties using a pair of client and server random values together with a pre-master secret key. The client and server random values are exchanged during the handshake, while the pre-master shared key is used or obtained depending on the authentication procedure, which fundamentally depends on the cipher suite employed. In particular, using cipher suites employing public-key authentication the client is allowed to generate the pre-master shared key and send it to the server encrypted with the server's public-key. Therefore, this is what happens when using the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 suite with the *Certificates* CoAP security mode. Pre-shared key suites as TLS_PSK_WITH_AES_128_CCM_8 [17] don't support this, because at an initial stage the two entities are unable to support the secure transmission of the pre-shared secret. As this single limitation would prevent end-to-end agreement of the pre-master secret key in the context of our proposed mediated DTLS authentication, we modify DTLS pre-shared key authentication using TLS_PSK_WITH_AES_128_CCM_8 to enable the 6LBR to transmit the pre-master secret to the CoAP server running on the sensing device. Thus, the pre-master secret key received from the Internet client is forwarded to the CoAP server and stored at the 6LBR if required for additional security mechanisms, as we discuss later. In order to guarantee security for the transmission of this secret in the LoWPAN, we define an associated authentication protocol supported by the CA, that we discuss later in the paper.

Returning to our analysis of the message exchange illustrated in Figure 3, the *ClientHello* message confirming the initial request also transports the client random value, the protocol version and the list of cipher suites supported by the client. After reception of this message, the 6LBR requests from the AC server security-related information concerning the destination CoAP sensing device, in particular its supported cipher suites and its X.509 certificate. This request is part of the LoWPAN authentication protocol as we describe later. The *ClientHello* message includes a request for public-key authentication and is forwarded by the 6LBR to the CoAP server with a request for pre-shared key-based authentication, as appropriate for TLS_PSK_WITH_AES_128_CCM_8. This is the currently evaluated cipher suite in our architecture, although other ciphers may be adopted in the future.

The *ServerHello* message containing the server's response is also forwarded back to the CoAP Internet client, with an acknowledgement for public-key authentication included in the message. The following *ServerKeyExchange* message contains the server random value and is also forwarded to the CoAP client, the same applying to the *ServerHelloDone* message terminating this message flight. In the following message flight the 6LBR authenticates the CoAP server on its behalf by sending the appropriate X.509 certificate previously received from the AC server. The 6LBR also requests that the client authenticates itself with its own certificate. This message flight finishes with the *ServerHelloDone* message. Next the client sends its certificate and a *ClientKeyExchange* message containing the client's random value and pre-master secret key generated by the client, which the 6LBR forwards to the sensing device supporting the CoAP server.

As we illustrate in Figure 3, pre-master secret key agreement is preceded by mutual authentication between the 6LBR and the CoAP server via the AC server, that we detail later in the context of the LoWPAN authentication protocol. After reception of the *ClientKeyExchange* message, both CoAP entities are in possession of the same pair of random values and pre-master secret key required to compute the DTLS master key, and from this key the secret material for DTLS security may be derived.

Authentication and PMSK exchange on the LoWPAN

As previously discussed, our architecture modifies TLS_PSK_WITH_AES_128_CCM_8 to support pre-master secret key exchange in the context of the handshake, more precisely by propagating this value towards the CoAP sensing device using the initial *ClientKeyExchange* message. One important goal of ours is not to compromise end-to-end security by accepting low security for messages exchanges on the LoWPAN, and as such we introduce an authentication protocol supported by the AC server with the goal of guaranteeing appropriate security for communications between the 6LBR and CoAP sensing devices. This authentication protocol is integrated with the two-phase DTLS handshake controlled by the 6LBR, and fulfills the important goal of guaranteeing a high-degree of security for end-to-end communications at all stages of an end-to-end DTLS session. Figure 4 illustrates the messages exchanged by the LoWPAN authentication protocol. This protocol supports confidentiality of the messages exchanged during the handshake and mutual authentication between the 6LBR and CoAP device, while assuming the AC server to be a trusted entity.

The proposed LoWPAN authentication protocol inherits characteristics from Kerberos [18], while introducing others required to support the two-phase delegated DTLS handshake and the transportation of the pre-master secret key. The AC server is responsible for maintaining security-related information for each registered LoWPAN CoAP device. In particular, for each device the AC stores its client ID, its X.509

ECC certificate and the list of supported ciphers and compression methods. The current mandatory cipher is TLS_PSK_WITH_AES_128_CCM_8, although further ciphers may be adopted in the future, as long as compatibility is maintained with the cipher employed for communications on the Internet domain. The certificate may be preconfigured for a sensing device or in alternative directly obtained from the CA server whenever required, as illustrated in Figure 2. Compression negotiation is supported by the DTLS handshake and also with the mediated DTLS handshake. The client ID for a CoAP device is its LoWPAN IPv6 link-local address. We assume that communications between the AC and 6LBR run over a communications medium without the limitations of the LoWPAN.

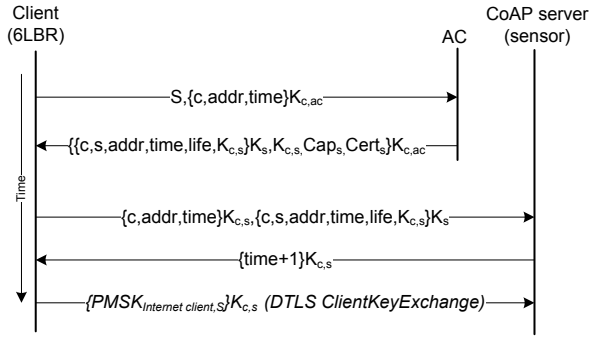

Figure 4. LoWPAN support authentication protocol

The 6LBR and AC server share a secret key ($K_{c,ac}$ in Figure 4) employed to encrypt messages exchanged between the two devices. The goal of the first message flight in the authentication protocol is to enable the 6LBR to obtain security-related information for the destination CoAP device. This information consists of its certificate, the list of supported encryption and compression methods and an access token for subsequent authentication of the 6LBR to the CoAP device. The request in the first message identifies the CoAP server device and the address of the 6LBR, while also including a timestamp. The AC server builds an authentication token with the previous information plus a lifetime value and the secret session key ($K_{c,s}$) to be used by the 6LBR and the CoAP server. The authentication token is encrypted with a secret key that the AC server shares with the CoAP device (K_s) and is forwarded unmodified by the 6LBR to the CoAP device. In this reply the 6LBR also receives the secret session key, a list of ciphers and compression methods supported by the CoAP device, and its public-key certificate. Depending on the ciphers supported by the CoAP device, the 6LBR may decide to terminate the two-phase handshake at this stage, and in consequence the DTLS handshake illustrated in Figure 3 would terminate by returning a *Finished* message to the Internet CoAP client.

The second message flight supports mutual authentication between the 6LBR and CoAP sensing device and the secure pre-master secret key exchange. The 6LBR transmits the authentication token previously obtained from the AC server together with a similar token containing its identification and address plus a timestamp. The CoAP server compares the information contained in the two tokens received in order to authenticate the 6LBR, while also analyzing the timestamp and lifetime values. These values offer protection against message replay attacks. In the case of successful authentication, the CoAP server is now in the possession of the secret session key $K_{c,s}$. The following reply message is encrypted with this key and authenticates the CoAP server to the 6LBR, by having the server transmit the received timestamp plus one. The final message is the *ClientKeyExchange* message sent in the context of the two-phase mutual DTLS handshake. This message transports the pre-master secret key and modifies the TLS_PSK_WITH_AES_128_CCM_8 suite as previously discussed. After this last message the DTLS handshake proceeds as previously illustrated in Figure 3. After the computation of the master secret and of the keying material on the CoAP client and server, end-to-end DTLS security may be enabled employing AES/CCM. AES/CCM may be supported in software on the Internet CoAP entity and (when available) by hardware cryptography on the sensing device.

V. EXPERIMENTAL EVALUATION

The mechanisms previously discussed in the context of the proposed architecture may contribute to the security of Internet-integrated LoWPANs and to the intelligent allocation of limited resources available on CoAP sensing devices to security. ECC public-key authentication and key negotiation as proposed for CoAP may be supported for Internet-integrated sensing applications using devices unable to otherwise support it directly. Also, attacks and threats due to the integration of LoWPAN communications and devices with the Internet may be efficiently circumvented using mechanisms deployed on a non-constrained 6LBR device.

Experimental evaluation setup

The system architecture illustrated in Figure 2 is implemented for experimental evaluation purposes, with the main goal of comparing the impact of end-to-end security as proposed in our architecture against the original proposal for CoAP security. We employ a TelosB [15] sensing device and Linux hosts, with the TelosB supporting the TinyOS [19] operating system with the Berkeley Low-IP (BLIP) 6LoWPAN stack, plus CoAP support and the two different DTLS configurations. We may note that, although the experimental results are specific to the TelosB, they may provide an acceptable reference considering the representativeness of the TelosB. The TelosB is powered by a 16-bit RISC MSP 430 microcontroller with 48Kbytes of ROM and 10Kbytes of RAM, supporting communications at 2.4GHz and data transmissions at 250Kbps. We support standalone AES/CCM encryption available in the TelosB using the encryption code from the Shanghai Jiao Tong University [20], while ECC is supported using code based on TinyECC [21]. The 6LBR, the CA server, the AC server and the Internet CoAP client are supported using Linux. The 6LBR supports routing between an Ethernet IPv6 network and the IEEE 802.15.4 LoWPAN by employing a second TelosB mote in

bridge mode. The Internet CoAP client is uses *libcoap* [22] integrated with DTLS support. The TelosB and the AC server support the LoWPAN authentication protocol.

Impact on the resources of constrained sensing devices

Our initial evaluation is on the RAM and ROM memory required to support end-to-end security, given its scarcity on sensing platforms such as the TelosB.

1. Memory footprint of end-to-end security

Our following discussion identifies the proposed end-to-end CoAP security mode as ME2ECoAP (mediated end-to-end CoAP security using the delegated handshake with mutual authentication), while the original end-to-end CoAP security mode is identified as E2ECoAP. In Figure 5 we illustrate the impact of end-to-end security on the memory of the TelosB, while also including the base usage scenario without security. The illustrated values are obtained considering the support of TinyOS with BLIP and CoAP plus the code required to support the appropriate cipher and DTLS security. We also consider the support of TLS 1.2 PRF using SHA-256, as required by CoAP to support integrity [5]. ME2ECoAP also includes the code required to support the LoWPAN authentication protocol.

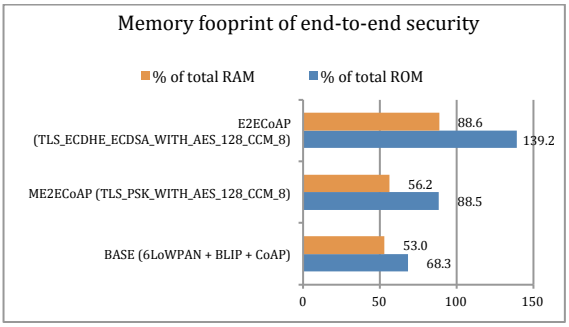

Figure 5. Memory footprint of transport-layer end-to-end security

We observe that hardware-level encryption doesn't come without a non-negligible overhead on memory, particularly ROM. The limitations of the TelosB in terms of memory are visible, as more ROM memory is required to fully support end-to-end security using the original CoAP *Certificates* security mode. RAM may also be a problem in usage scenarios where larger applications require more available memory from the sensing device, the same also applying to the storage and processing of X.509 certificates and related public-keys. In general, we may observe the superior performance of ME2ECoAP in terms of memory usage and availability using the TelosB to support the CoAP server.

2. Expected lifetime of sensing applications

Energy is certainly another scarce resource in constrained sensing platforms, and many sensing applications must be designed with battery-powered sensing devices in mind and to run for acceptably long periods of time. In order to obtain the expected lifetime of IoT sensing applications employing end-to-end security we start by experimentally measuring the impact of packet processing, security and communications on the energy available on the TelosB. Energy was obtained using experimental measurements of the voltage across a current resistor placed in series with the battery pack of the TelosB. In particular, we measure the energy required to support the DTLS handshake (handshake processing plus handshake communications energy) and the energy required to support DTLS encryption using AES/CCM (DTLS encryption plus communications energy). For all measurements we consider the usage of 6LoWPAN 102-bytes packets as previously discussed in the context of Figure 1. Regarding the handshake, the original DTLS handshake requires a total of 39 6LoWPAN 102-bytes messages and a total of 54.4 mJ (Millijoules), in contrast with our delegated two-way handshake, which requires 15 LoWPAN messages (including messages of the LoWPAN authentication protocol) and 0.001 mJ. Regarding DTLS encryption, 0.0002 mJ are required to process security for packet using AES/CCM and 10.89 mJ for digital signing using ECC as required for TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8. We may clearly observe that ECC-based cryptography represents a bottleneck using the TelosB. These values are total, measured from the reception of a 6LoWPAN packet to the time when cryptography finished processing the packet, and thus represents the total energetic effort to process end-to-end security for a packet. Finally, the energy required for the processing of a packet and related security headers was measured as 0.007 nJ (Nanojoules) and is accounted for in our following evaluation. From the experimental values previously discussed we derive the expected lifetime for a sensing application, which we illustrate in Figures 6 and 7.

The expected lifetime values illustrated in Figures 6 and 7 considers the usage of the TelosB sensing device powered using two new AA LR-6 batteries and applications with different requirements in terms of the number of DTLS sessions established per hour and the number of CoAP requests served per DTLS session. We count a CoAP request as two 102-bytes 6LoWPAN packets, one containing a confirmable request and the other its reply.

In Figure 7 we observe again the superior performance of ME2ECoAP, given that in the worst scenario (corresponding as illustrated to 19 DTLS sessions per hour with 10 CoAP requests per session) the expected lifetime is about 29900 hours, approximately 5 times the corresponding value for E2ECoAP (5461 hours). We may also observe a more expressive decline for ME2ECoAP in respect with the expected lifetime when the number of CoAP requests per session increases. This is due to the larger impact of AES/CCM security in comparison with the impact of the DTLS handshake, in contrast with E2ECoAP in Figure 7 for which the lifetime is dominated by the much larger impact of the DTLS handshake. Despite this, in all usage scenarios ME2ECoAP is superior in respect to the expected lifetime.

Overall, ME2ECoAP would be the best choice for sensing applications designed to operate in a closed fashion, where

CoAP devices are able to maintain security sessions with a closed set of Internet devices for long time periods, but also for open applications where CoAP devices accept requests from any Internet client.

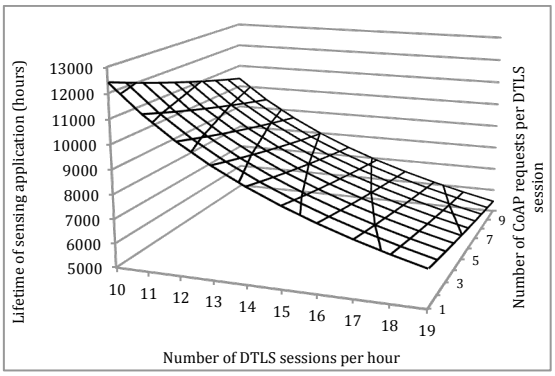

Figure 6. Impact of end-to-end security on the lifetime of sensing applications (E2ECoAP)

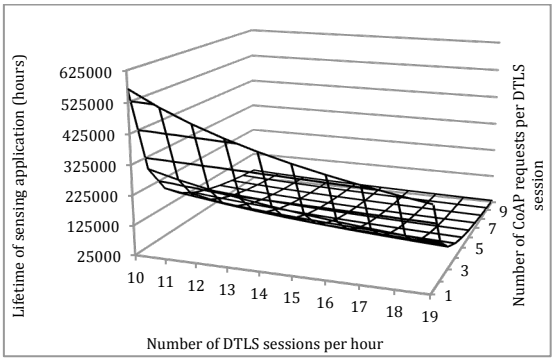

Figure 7. Impact of end-to-end security on the lifetime of sensing applications (ME2ECoAP)

3. CoAP communications rate

As advanced mechanisms such as multi-threading are usually absent from low-end microcontrollers such as the MSP430, the computational time required to support security directly influences the maximum communications rate that a sensing device may support. We experimentally measure the computational time required to support the DTLS handshake (handshake processing plus handshake communications delay). The DTLS handshake employing the original CoAP security proposal requires 10.09s, in depth contrast with the DTLS delegated handshake, which requires 15.39ms. Such values include the time required for communications in the two handshakes, and for the later also the time required for the LoWPAN authentication protocol. This clear difference is again due to the large impact of ECC cryptography on the TelosB, giving that ECC digital signing is required to process a few of the messages of the handshake. ECC encryption for digital signing requires a total of 2019.6 ms, while with ME2ECoAP this is not an issue since ECC computation is delegated to the 6LBR proxy. We again consider the overhead of AES/CCM, which is of 3.6ms per packet.

Based on the experimentally obtained values previously discussed, we derive the maximum number of CoAP requests that a CoAP sensing device may support with end-to-end security, which we illustrate in Figure 8. The illustrated values reflect the weight of the DTLS handshake in the overall CoAP communications rate. We may observe that, although the difference in the performance of the two end-to-end security modes may be of less significance for applications requiring a smaller number of DTLS sessions per hour, for others ME2ECoAP is clearly the best choice.

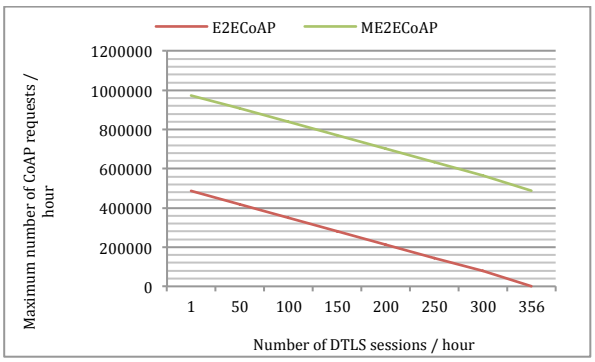

Figure 8. Impact of the DTLS handshake on the available time to process CoAP requests (experimental values for the TelosB)

In particular, we observe that the original DTLS handshake is only viable up to around 356 DTLS sessions per hour (roughly one new session each 10 seconds), due to the computational weight of ECC. For applications requiring a larger numbers of secure sessions per hour the current proposal for CoAP security is completely unviable. Regardless of the number of DTLS sessions per hour required by a particular sensing application, ME2ECoAP is clearly the most appropriate choice.

VI. Conclusions and Future Work

We propose an end-to-end security architecture for Internet-integrated sensing applications providing benefits not only in respect to the efficient support of ECC authentication and key agreement, but also of other mechanisms promoting security of LoWPAN devices and communications. As verified with our experimental evaluation, when employing current sensing platforms the delegation of costly ECC computations to a more powerful device clearly pays off, even with the additional overhead of supporting an additional LoWPAN authentication protocol. As future work, additional security mechanisms may be supported by the 6LBR in the context of the proposed architecture. For example, the analysis of encrypted CoAP communications at the 6LBR may support detection of attacks at the application-layer, at the end also contributing to security in the context of Internet-integrated sensing applications.

Acknowledgment

The work presented in this paper was partly financed by the iCIS project (CENTRO-07-ST24-FEDER-002003), which is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER.

REFERENCES

[1] Kushalnagar N et al. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. *RFC 4919*, 2007.

[2] Montenegro G et al. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. *RFC 4944*, 2007.

[3] Hui J et al. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. *RFC 6282*, 2011.

[4] Shelby Z et al. Constrained Application Protocol (CoAP). draft-ietf-core-coap-13, 2012.

[5] Shelby Z. Constrained RESTful Environment (CoRE) Link Format. *RFC 6690*, 2012.

[6] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE std. 802.15.4, 2006.

[7] Chen X, Makki K, Yen K, Pissinou N. Sensor network security: a survey. *IEEE Communications Surveys & Tutorials* 2009; 11(2), pp. 52-73, DOI: 10.1109/SURV.2009.090205.

[8] Rescorla E et al. Datagram Transport Layer Security Version 1.2. *RFC 6347*, 2012.

[9] Luk M et al. MiniSec: A Secure Sensor Network Communication Architecture. *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN 2007)*, Cambridge, Massachutts, USA, 2007. DOI: 10.1109/IPSN.2007.4379708.

[10] Gupta V et al. Sizzle: a standards-based end-to-end security architecture for the embedded Internet. *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, Kauai Island, HI, USA, 2005. DOI: 10.1109/PERCOM.2005.41.

[11] Jung et al. SSL-based Lightweight Security of IP-based Wireless Sensor Networks. *Proceedings of the International Conference on Advanced Information Networking and Applications Workshop (WAINA '09)*, Bradford, UK, 2009.

[12] Raza S et al. 6LoWPAN Compressed DTLS for CoAP. *Proceedings of the IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS 2012),* Hangzhou, China, 2012. DOI: 10.1109/DCOSS.2012.55.

[13] Kothmayr T et al. A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-Way Authentication. *Proceedings of the Seventh IEEE International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp 2012),* Clearwater, FL, USA, 2012.

[14] Granjal J et al. On the Effectiveness of End-to-end Security for Internet-integrated Sensing Applications. *Proceedings of The IEEE International Conference on Internet of Things (iThings 2012)*, Besançon, France, 2012. DOI: TBD.

[15] TelosB Mote Platform, http://www.xbow.com/pdf/Telos_PR.pdf (accessed Jan 14 2013).

[16] SECG-Elliptic Curve Cryptography-SEC 1, http://www.secg.org (accessed Jan 14 2013).

[17] Eronen P, Tschofenig H. Pre-Shared Key Ciphersuites for Transport Layer Security (TLS). *RFC 4279*, 2005.

[18] Neuman B, Ts'o T. Kerberos: an authentication service for computer networks. *IEEE Communications Magazine,* 1994, 32(9), pp. 33-38, DOI: 10.1109/35.312841.

[19] TinyOS Operating System, http://www.tinyos.net/ (accessed Jan 2013).

[20] Standalone hardware AES Encryption using CC2420, http://cis.sjtu.edu.cn/index.php/The_Standalone_AES_Encryption_of_CC2420_(TinyOS_2.10_and_MICAz (accessed July 2012).

[21] Liu A., Ning P. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. Proceedings of the 7th h international conference on Information processing in sensor networks (IPSN '08), 2008

[22] LibCoAP, http://sourceforge.net/projects/libcoap/ (accessed Jan 2013).